

Application for United States Patent

of

Heather Maria Hinton, *et al.*

for

5 “System and Method for User Enrollment in an e-Community”

CROSS-REFERENCE TO RELATED APPLICATIONS

(CLAIMING BENEFIT UNDER 35 U.S.C. 120)

This application is related to US Patent Application number _____ (to
be amended to include USPTO serial number), docket number

10 AUS9-2000-0770-US1, filed on November 9, 2000, by Heather M. Hinton.

FEDERALLY SPONSORED RESEARCH

AND DEVELOPMENT STATEMENT

This invention was not developed in conjunction with any Federally sponsored
contract.

15 MICROFICHE APPENDIX

Not applicable.

INCORPORATION BY REFERENCE

The related US Patent Application number _____, docket number
AUS9-2000-0770-US1, filed on November 9, 2000, by Heather M. Hinton, which is

commonly assigned is hereby incorporated by reference in its entirety, including figures and Information Disclosure Statement.

Patent Application

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] This invention relates to online user identification, authentication, and authorization systems and methods, and especially to cross-domain log on technologies and technologies which create and manage virtual communities of online users.

Description of the Related Art

[0002] Each Internet user is served by a "home domain", which is a domain in which a user is "registered". A user typically "logs in" to his or her home domain using a user ID or name and password. Then, after successfully completing a log in process, the user is allowed to access secured information and resources within the home domain to which the user is entitled to access or use according to the user's account definition. The user, then, has a 'long term relationship' with his or her home domain.

[0003] In addition, the home domain itself may have 'long term' relationships with other domains. For example, a search engine web site provider may maintain a long term relationship with a service provider, such as an online insurance quote provider. This is a typical characteristic especially for business-to-business ("B2B") and e-community domains, where one domain (e.g. the home domain) is responsible for user registration issues, including such issues as help desk support and password management.

[0004] Often, a user will access resources in different ("participating") domains on behalf of their home domain. In some instances, the user will have to resubmit to a log in or authentication process as he or she moves from the home domain to another domain.

- 5 [0005] To address this problem, the related patent application described a method to allow a user to transfer to another participating, secure domain without having to re-authenticate to this second domain. This process was referred to as "cross-domain single-sign-on".

- 10 [0006] The drawback with the method described in the related patent application is that a user can only transfer to a participating domain directly from the user's home domain, and not across from one participating domain to another participating domain. While being of some usefulness to the user, this process effectively requires the user to return to the home domain before proceeding to another participating domain rather than going directly to the other participating domain.

- 15 [0007] Still other available solutions to this problem do not allow for a "long term" relationship with a domain that is not the home domain in which a user is registered or initially authenticates. These other solutions require a user to transfer to a new domain via the user's authenticating domain, usually by triggering a hypertext transfer protocol ("HTTP") redirection to the new domain.

- 20 [0008] Therefore, there is a need in the art for a cross-domain single-sign-on system and method which allows an Internet user to establish a long-term relationship with participating domains, and which gives the user the ability to go directly to

participating domains, via bookmarks or direct URL's for example, without having to go through a home domain first. Further, there is a need in the art for this new system and method to provide a simple user experience wherein the user does not need to know anything about the e-community in which he or she is participating.

- 5 Another advantage of the approach proposed in this invention is that it is easy to implement, easy to use, and provides a secure method of cross-domain single-sign-on functionality.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The following detailed description when taken in conjunction with the figures presented herein provide a complete disclosure of the invention.

[0010] Figure 1 illustrates a simple e-community architecture.

5 [0011] Figure 2 shows a first portion of the logical process of the invention for a user to enroll in a specific domain within an e-community.

[0012] Figure 3 shows a second portion of the logical process of the invention for a user to enroll in a specific domain within an e-community.

[0013] Figure 4 sets forth the high-level logical process according to the invention
10 for registering, enrolling, authenticating, vouching for, authorizing, participating, and disengaging to and from an e-community by a user.

[0014] Figure 5 shows a first portion of the logical process of the invention for a user to enroll in a group of participating domains within an e-community.

[0015] Figure 6 shows a second portion of the logical process of the invention for a
15 user to enroll in a group of participating domains within an e-community.

SUMMARY OF THE INVENTION

[0016] The present invention allows an Internet user to transfer directly to a domain that is participating in the e-community, by means such as a Bookmark or a directly-typed URL, without the necessity of returning to a home domain prior to transferring to the participating domain. This enhances the usability of the e-community and set of participating domains, and allows the user to build a long-term relationship with multiple participating domains, all within the scope of an e-community.

[0017] Initially, a user registers in a "home domain", where the user establishes a valid user account and password. Subsequently, a user may first authenticate (e.g. log-in to) a home domain, which creates and holds a credential for the user.

[0018] Later, the user may attempt to access or use resources in another domain affiliated with the e-community from the user's home domain which results in the home domain sending an enrollment request message to the affiliated domain with an enrollment request token. The enrollment request is sent via the user's browser using HTTP redirection. Included in this request is a home domain identity cookie (DIDC) set by the home domain. The user's browser extracts and stores the home DIDC.

[0019] The affiliated site receives the enrollment request message, unpacks the enrollment request token, and builds an affiliated domain identity cookie (affiliated DIDC) for that user. An enrollment success message is sent by the affiliated domain to the home domain, including the affiliated DIDC and a success indicator. Again, the

message is sent via the user's browser using redirection. The user's browser extracts and stores the affiliated DIDC.

[0020] The home domain receives the enrollment success message and modifies the home DIDC to include a "enrolled at affiliated domain" token in a portion of
5 extensible data in the home DIDC. This updated home DIDC is then transmitted to the user's browser, where it is stored in the persistent cookie store.

[0021] Enrollment into other affiliated domains may proceed similarly until the user has enrolled in all user-selected affiliated domains, or has been automatically enrolled in all domains belonging to a group of affiliated domains. This enrollment may be
10 explicitly initiated by a user for a single domain or for a group of affiliated domains. The enrollment process may also be automatically (e.g. "under the covers") triggered based on a user's actions.

DETAILED DESCRIPTION OF THE INVENTION

[0022] The present invention extends or builds upon the invention of the related patent application. As such, the web-based cross-domain single-sign-on ("CD-SSO") authentication system and method of the related patent application forms a part of the preferred embodiment of the present invention. It will be recognized, though, by those skilled in the art that the present invention may be utilized and realized in conjunction with other user authentication technologies and processes.

[0023] In particular, an "introductory authentication token", as described in the related patent application, is an integral part of the present invention. For the purposes of the present disclosure, the introductory authentication token will be referred to as a "vouch-for" token.

[0024] An "enrollment" function is added to the process of the related patent application, and the vouch-for token is expanded to include extensible metadata. The present invention does not require an SSL connection to be established for the transfer of the vouch-for token.

[0025] Further according to the preferred embodiment, this disclosure presents the e-community single-sign-on functionality as disclosed in the related and incorporated patent application. This functionality can be implemented as a part of an existing security policy management product, as a "plug-in" to an existing Web proxy server, such as Netscape or Microsoft IIS, or as a plug-in to an existing Web application server, such as IBM WebSphere Application Server. It will be readily recognized by those skilled in the art that implementation with other web services software and

platforms is possible without departing from the spirit and scope of the present invention.

[0026] The present invention is designed with the requirements for global corporations and for “e-communities”, such as B2B relationships, in mind. These

5 requirements are for:

- a. Single-sign on functionality between e-community members within a DNS domain.
- b. Single-sign on functionality between e-community members across DNS domains (such as www.ibm.com and www.acme.com).
- 10 c. Co-existence with existing infrastructure: this solution can be implemented as a plug-in, allowing plugging it in to a wide variety of front end Web Servers.
- d. Extensible data included in enrollment tokens to be used for personalization and branding purposes (not for security purposes).

15 [0027] The goals met by the preferred embodiment of the present invention are:

- a. to provide a Cross-domain single-sign-on functionality that can work with existing systems with minimal software and hardware modifications/additions;
- b. realize a CD-SSO functionality that does not require a “master authentication” server
- 20 c. provide a CD-SSO solution that is scalable to tens of millions of users;

- d. provide a CD-SSO functionality with the ability to transfer “meta-data” such as is used for branding or co-branding purposes;
- e. provide a CD-SSO functionality which fits within the “extranet” space.

Process Participants

5 [0028] According to the present invention, an "e-community" has many different "participants", including e-community members, or domains corresponding to the business units that are participating in the e-community, and users who are employees or clients of the e-community members. In general, a user is an employee of one and only one e-community member, so this e-community member represents the user's
10 home domain.

[0029] Within the e-community, there are specific roles that must be filled. An e-community “owner-member” is the e-community member that maintains common functionality required by all e-community members. An example of this type of functionality would be enrolling new members into the e-community.

15 [0030] “Key administrators” are able to invoke the key management functionality at the e-community owner-member, where each e-community member will have one or more key administrators.

[0031] “User administrators” may invoke limited user administration functionality at other e-community members. This user administration functionality is preferably
20 limited to user creation, deletion and identity mapping for users in the “e-community”

groups only. This allows each e-community member to manage their own users and to have limited responsibility for other e-community member's users.

[0032] An e-community has, in general, more than two participants. Figure 1

illustrates a simple e-community architecture, where a user (100) accesses the

5 e-community from their browser. In this example, there are three participants in the e-community: the user's home domain (103), an "other" domain (106) and "another" domain (108).

[0033] Within the home domain (103), there are two web "front-ends". The

e-community functionality is preferably implemented as part of these front-ends as a

10 e-Community SSO plug-in (109). The front-ends (101, 102, 104, 105, and 107) may represent "clusters" (e.g. a set of replicas), or they may represent single machines.

Process Phases

[0034] The process is designed to handle all combinations of in-domain and

out-of-domain requests to protected as well as unprotected resources by a user that is

15 or is not already authenticated into the CD-SSO domain. The CD-SSO process (40) contains the following steps, as shown in Figure 4:

a. registration of the user into the e-community (a prerequisite of the protocol) (41), and/or registration of a member (or domain) into the e-community (41');

20 b. enrollment of the user into the e-community (42) and or enrollment of a member (or domain) into the e-community (42');

- c. authentication (43) of user into the user's "home" domain within the e-community;
- d. subsequently vouching (44) for the user's authenticated identity by the home domain to a participating e-community domain;
- 5 e. authorization (45) of the user in the e-community;
- f. participation (48) in e-community (e.g. user accesses or uses resources of domains in e-community)
- g. logging out (46) of user from the e-community; and
- i. removal (47) of the member from the e-community.

10 [0035] At any time during the process (40), if the process is deemed to have "failed", that is, any of the authentication verification tasks fail, then an "access denied" message is returned (48) to the user and the process stops.

Protected versus Plaintext Versions of Process

15 [0036] According to the preferred embodiment, this process is intended to be used with "protection" on the tokens used to transfer and maintain state information. This protection is in the form of encryption for confidentiality, and hashes for integrity.

[0037] For those installations that use this e-community single-sign-on functionality for personalization purposes and which do not have strong security requirements, this encryption and hashing may be omitted. This disclosure describes the secure version
20 of this process.

User Perspective of e-Community Enrollment Process in General

[0038] Participants in the e-community must be “registered” into the domain. This means that the participant must have a valid user identify in at least one of the member domains that is within the e-community.

5 [0039] There is one and only one domain that is defined as a user’s home domain. For example, an employee of IBM would have www.ibm.com as his or her home domain when participating in an e-community relationship with ACME, REMCO, and BigCo.com.

10 [0040] Registration into the e-community home domain is not a part of this process -- it is a requirement for this functionality. Each member of the e-community must have some means of mapping the identity of a user from another domain into an identity that is valid within that member’s domain.

[0041] The e-community solution provides an exit so that each e-community member can provide their own identity mapping procedure. This functionality will be
15 established when enrolling a member into the e-community.

[0042] User enrollment into the e-community is the act of establishing a relationship between a user and the e-community members. Enrollment requires that a user have a valid account established within the user’s home domain. As a result of enrollment into the e-community, a user will have a “domain identity cookie” ("DIDC")
20 established by each of the participating e-community domains. This domain identity

cookie will be used by the individual e-community members when implementing the single-sign-on functionality.

[0043] The purpose of the domain identity cookie is to identify the user's "home" domain, to identify a URL in the user's home domain that can "vouch for" the user's identify, and to identify the e-community in which this user is a participant. The DIDC can also be used for personalization purposes, such as branding or co-branding based on the user's home domain. The DIDC may also be used to distinguish which e-communities of multiple e-communities to which a user belongs.

[0044] A user enrolls in the e-community through his or her home domain. As a prerequisite for enrolling in the e-community, the user is required to authenticate to his or her home domain. This allows for an access control decision to determine if the user has "e-community" privileges. Thus, not all of a company's employees, for example, will be allowed to participate in an e-community relationship.

[0045] Enrollment in the e-community may require an explicit action on the part of the user, or it may be implemented as an "under-the-covers" automatic enrollment functionality triggered by the user's actions. Once this process has been initiated, it is the responsibility of the user's home domain to enroll the user in all participating e-community domains.

[0046] There are two ways that enrollment can be accomplished, according to the preferred embodiment. The first method of enrollment enrolls the user at a group of affiliated sites; this is referred to as "group enrollment". This group corresponds to

the minimal set of affiliate sites in which all users must be enrolled. For example, this may correspond to a set of distinct DNS named resources for a single corporation.

[0047] The second method of enrollment enrolls a user at a specific individual site; this is referred to as “individual enrollment”. This method of enrollment is activated
5 through an “enroll at X” hyperlink that in turns invokes the enrollment functionality to enroll a user in domain "X" only.

Group Enrollment Process

[0048] During group enrollment, a user is enrolled in a set of domains that make up a “minimal” set of domains within the e-community. This group may be defined by the
10 user’s home domain, and may correspond to the common set of domains required by all users participating in a given e-community.

[0049] When a user activates the group enrollment functionality, the user is preferably redirected from one domain to another, until each domain has been visited once. The user’s home domain is responsible for redirecting the user to each of the
15 participating domains for the purpose of enrolling in the e-community following a “star” topology. This allows the home domain to determine and report the status of each enrollment attempt across the e-community.

[0050] This functionality may be implemented as part of an “e-community portal enrollment” resource. That is, each participant within the e-community will provide an
20 “enrollment” page with the resources required to trigger the enrollment functionality.

The functionality behind the “enroll” resource is responsible for re-directing a response through the user to every participating domain in the e-community in turn.

[0051] The group enrollment process (50) is shown in Figure 5, based on the components shown in Figure 1. This example assumes that the user has not previously
5 enrolled in the e-community, and that a user does not have any domain identity cookies from any of the e-community members.

[0052] First, the user (100) accesses (51) an “enroll in e-community” resource at domain (103), at which time the SSO plug-in (109) at home domain (101) receives
10 (52) this request and checks (53) if the user (100) has authenticated to the home domain (101).

[0053] If the user has not already authenticated to the home domain, the SSO plug-in (109) at the home domain prompts (54) user (100) for authentication information, and performs (55) authentication verification.

[0054] Once the user has been authenticated, an access control decision is made to
15 determine if the user is able to enroll in the e-community (56). According to the preferred embodiment, this authorization check is performed by the security policy server authorization engine.

[0055] If the user is not entitled to enroll in the e-community, the process stops (57) with an error message issued to the user. If the user is permitted to enroll in the
20 e-community, the process continues with the SSO plug-in (103) at the home domain building (58) a single-sign-on cookie, e-Community cookie, such as:

eCC(101) = {Auth Server = 101, URI at Auth Server = www.103.com/101/
vouch_for.htm, e-community = sample, creation date = Nov 1, 2000,
extensible attribute=value pairs}, hash(info)

where the extensible data is configured by member "103."

5 **[0056]** Next, the SSO plug-in (103) consults (59) an internal resource list to
determine who are the participants in the e-community. This list contains the other
domain (106) and another domain (108).

10 **[0057]** The plug-in (109) then builds an identity cookie DIDC (103) and an
"enrollment token" for the user (100), and creates a response, re-directed to the other
community domain (106). As previously discussed, the enrollment token can be sent
"in the clear" or cryptographically protected.

15 **[0058]** Continuing to Figure 6, the user's browser extracts (62) the DIDC and stores
it in the browser's persistent cookie store, such as storing it in a cookie folder on a
hard disk drive, and redirects the response to the other community domain (106). The
plug-in (109) at the other domain (106) front-end (104) receives (63) the enrollment
request from the home domain front-end (101) which was redirected through the user
(100).

20 **[0059]** The plug-in (109) at the other domain's front end (104) "unpacks" the
enrollment token, and builds an domain identity cookie for the user for the other
domain (106). An "enrollment successful" message is then sent to the home domain's

front-end (101) via redirection (63) through the user's browser (100) along with the domain identity cookie for the other domain (106). During redirection (64) at the user's browser, the user's browser extracts (64) the DIDC for the other domain (106) and puts it in the browser's persistent cookie store.

- 5 **[0060]** Finally, the home domain (103) plug-in (109) at the first front-end (101) receives (65) the redirected "enrollment successful at other domain" message. The SSO plug-in (103) at the first front-end (101) modifies (65) the home domain DIDC to include an "enrollment success at other domain" symbol in the extensible attribute data. This modified cookie is then returned (65) to the user in the next response from
10 the first front-end (101).

- [0061]** In this manner, the home domain DIDC is "built up" or accumulated to include indicators of successful enrollments at affiliated domains within the e-community. This process may continue for additional domains in the e-community, using the user's browser as a re-direction node in the communication path to pass
15 enrollment tokens and success tokens between the home domain and the affiliated domains, as shown in the remaining steps of Figure 6 (66 - 602).

- [0062]** As a result of this process, the user's browser receives a persistent domain identity cookie set by each of the e-community members (103, 106, 108) in which the user has successfully been enrolled. According to the preferred embodiment, the
20 cookie format is:

$$\text{DIDC}(x) = \{\text{home domain} = 103,$$
$$\text{vouch for URI} = \text{www.103.com/101/vouch_for.htm},$$

e-community=sample,

creation date = Nov 1, 2000,

extensible data(x) = data(x)}, hash(info)

[0063] Further, the user may be fully enrolled or partially enrolled. The behavior of
 5 the home domain is preferably configurable in the case when enrollment fails at an
 affiliated domain, such that the home domain plug-in can re-attempt enrollment at an
 affiliated domain until it is successful, or it can report an error to the user (and the
 e-community administrator).

[0064] Additionally, the user's browser now has a memory-based single-sign-on
 10 cookie set by the front-end that authenticated the user, in this case the first front-end
 (101) at the user's home domain. According to the preferred embodiment, the SSO
 cookie has the format:

eCC(101) = {Auth Server = 101, URI at Auth Server =
 www.103.com/101/vouch_for.htm, e-community = sample,
 15 creation date = Nov 1, 2000, extensible attribute-value pairs}, hash(info)

Individual Affiliate Site Enrollment Process

[0065] Enrollment of a user at an individual affiliate site enrollment is the process of
 enrolling a user at one participating e-community domain. It is a degenerate case of
 group enrollment, where the group size is just one. Enrollment at an individual

affiliate site typically occurs when a user activates an “enroll at site X” link at the e-community home domain, or if a user requests a link to a resource at site X from the home domain.

[0066] Thus, the e-community home is designed to maintain an “enrollment page”

5 with a list of all the individual sites at which a user may chose to enroll. Because different users will chose to enroll at different affiliate sites, each time a user invokes the individual affiliate site enrollment, the user’s domain identity cookie at the e-community home is updated to include the identity of the site at which the user has enrolled.

10 [0067] The user’s DIDC, as set by the e-community home domain, maintains a list of all the affiliate sites at which the user has explicitly enrolled. In order to protect this list should a user purge their cookie cache, a copy of this list may also be kept at the server side, preferably in the user’s lightweight directory access protocol ("LDAP") record so that the complete identity cookie can be rebuilt if necessary.

15 [0068] The individual affiliate enrollment process occurs as follows, again based on the components shown in Figure 1, where the user is enrolling in site X (e.g. site X could be the "other domain", for example). In this example, we also assume that the user has already enrolled in the e-community, and that the user has an existing DIDC from their e-community home domain.

20 [0069] First, the user (100) accesses (21) an “enroll at site X” resource which is preferably provided by his or her home domain (103). The SSO plug-in (109) at the

first front-end (101) of the home domain receives (22) this request, and checks (23) if user (100) has already been authenticated to the home domain front-end (101).

[0070] If the user has not already been authenticated to the home domain, then the SSO plug-in (109) prompts (24) the user (100) for authentication information (e.g.

5 user name and password), and performs (25) authentication verification.

[0071] Once the user is authenticated (or if the user has already been authenticated), the SSO plug-in (109) at the home domain front-end (101) builds (26) a single-sign-on e-Community cookie such as:

10 eCC(101) = {Auth Server = home domain(101),
 URI at Auth Server = www.103.com/101/vouch_for.htm,
 e-community = sample, creation date = Nov 1, 2000,
 extensible attribute=value pairs}, hash(info)

 where the extensible data is configured by the authenticating e-community member, which is in this example the home domain (103).

15 **[0072]** Next, an access control decision is made (27) to determine if the user is able to enroll in the e-community at the affiliated site. If not, the process stops with an error message to the user (28), otherwise, the process continues.

[0073] An “enrollment token” for the affiliate site is built (29) by the SSO plug-in (109) at the home domain front-end (101), and sent to the affiliate site via redirection
20 through the user's browser (100). As with the group enrollment process, the enrollment token can be sent “in the clear” or cryptographically protected.

[0074] As with the group enrollment process, the user's browser (100) extracts and stores (31) the cookie, and redirects (e.g. forwards) the message with the cookie to the affiliate domain.

5 [0075] The SSO plug-in (109) at the affiliate domain (106) front-end (104) receives the enrollment request, "unpacks" the enrollment token, builds a domain identity cookie for the user for the affiliate site, and returns (32) the DIDC with the changes to the home domain via redirection through the user's browser (100).

10 [0076] Similarly to the group enrollment process previously described, the user's browser receives (33) the DIDC from the affiliate site, extracts and stores a copy of the DIDC in persistent cookie storage, and forwards the message with the DIDC to the home domain, where it is received (34) by the SSO plug-in (109) at the home domain front-end (101).

15 [0077] Upon receipt of the DIDC from the affiliate domain, the home domain SSO plug-in modifies the DIDC to include an "enrollment success at affiliate site X" symbol in the extensible attribute data. This cookie is then returned to the user's browser in the next response from the home domain.

[0078] As with the group enrollment process, the DIDC is built-up or accumulated to include the enrollment status of the user with respect to the individual affiliate site, such as:

20 DIDC(x) = {home domain = 103,
vouch for URI = www.103.com/101/vouch_for.htm,
e-community = sample, creation date = Nov 1, 2000,

extensible data(x) = data(x)}, hash(info)

[0079] According to the preferred embodiment, the behavior of the SSO plug-in at the home domain is configurable to allow automatic retries to enroll at the affiliate site if the initial attempt fails, and/or to include notification to an administrator of the attempted and failed enrollment at the affiliate site.

Authentication of a User into their Home Domain in the e-Community

[0080] User enrollment in the e-community does not correspond to authentication into the e-community. Enrollment into the e-community is intended to initiate a long-term relationship between the user and the e-community domains.

10 Authenticating into the e-community is required to initiate a short-term relationship (e.g. a single session) such that the user can engage in e-community actions.

[0081] According to the preferred embodiment, the user can authenticate only to his or her home domain, which may be accomplished by explicitly logging into their home domain or by requesting a resource either within the home domain or in a domain

15 within the e-community in which the user has enrolled that requires an access control decision and therefore user authentication. These types of authentication processes are well known in the art, and many suitable authentication products and servers are available for this purpose.

[0082] Therefore, in one advantage of the present invention, our e-community
20 single-sign-on process does not require any changes to existing authentication

procedures. Successful authentication, will, however, results in an “e-community” cookie being placed at the user side.

[0083] If a user requests a protected resource in another domain, then authentication information must be transferred across the e-community. A method for accomplishing this is described later in this disclosure. Still, according to the preferred embodiment, the actual authentication of a user can happen in only one place -- their home domain.

[0084] As a result of authentication, the SSO plug-in generates an “e-Community Cookie” (an eCC or e-community cookie) that acts as an “authenticator bookmark”. This cookie is a memory cookie that is valid within the DNS domain, and will therefore be sent to any other server within the DNS domain.

[0085] The eCC identifies the server that authenticated the user and a URI pointing to an authentication script that can vouch for the user within a given domain.

[0086] Only the one instance within a DNS domain that authenticates the user or first receives an authentication “vouch-for” message sets an e-community cookie at the user’s browser. As such, a user has one e-community cookie set for each domain at which it has a current, authenticated (or vouched-for) session.

[0087] The e-community cookie indicates the security server or other plug-in location, and a URI at a plug-in location that can provide an authentication “vouch for” token for that user. This cookie is a domain cookie and can therefore be sent to any server in the domain that created it. This allows for simplified single-sign-on capabilities within a domain that is partitioned by multiple security server domains.

[0088] As an example, consider a hypothetical site "www.acme.com" that is partitioned so that there is a distinct security policy server set of replicas protecting each of the engineering, accounting, and human resource departments. In this situation, if a user authenticates (or is vouched for) first to engineering, they will have

5 a domain-wide e-community cookie set by the engineering security policy server.

When this user then goes to the accounting server, this e-community cookie indicates that the user has a current, authenticated session, and that the accounting server need not re-authenticate the user. Instead, the replica will request a voice-for token from the authenticating server as indicated in the eCC.

10 Vouching for User Identity across the e-Community

[0089] This is the process was described in the related patent application, and is utilized by the present invention. The process of vouching for a user's identity is sometimes referred to "transferring of authentication information" across the e-community. The implementation of the preferred embodiment does not transfer

15 authentication information, however. Instead, the user's home domain vouches for the identity of a user to another domain. This means that each member is responsible for managing the users in their domain and for providing a rule set of mapping the vouched-for identities from other domains.

[0090] The vouch-for process occurs when a non-home front-end receives a request

20 from a user that includes a domain identity cookie (DIDC) but not an e-community cookie (ECC) generated by that front-end. There are several steps to this process:

- (1) Identification that user is in the e-community but has a different home domain;
 - (2) Requesting (via re-direction) the user's home domain to "vouch for" the user;
 - 5 (3) Authentication of the user at the home domain (if not already authenticated);
 - (4) Generation of a "vouch for token" (VT) to transfer back to the requesting domain a re-directed response; and,
 - (5) Parsing of VT and creating of valid session for the user.
- 10 [0091] A prerequisite for the transfer of authentication information across domains is that the user has already enrolled in the e-community. If there is no DIDC cookie, the front-end will treat the user as a "normal" internal user (as opposed to a participant in the e-community) and will attempt to authenticate the user.
- [0092] Information is passed from the home domain to other domains in the
- 15 e-community through a "vouch for token," or VT. The VT is used to vouch for the authenticity of the user's identity to the other e-community domains. The VT will be created for each e-community domain only when requested and cannot be used by any e-community domain other than the intended domain. The VT is "transitory" in that it exists for the re-direction only and will not reside, ever, in the user's persistent or
- 20 memory cookie stores. The VT is protected by encryption, as well.

[0093] The VT is included in a response that is redirected back to the “requesting” e-community domain. When the requesting front-end/domain receives the response, it parses the VT, maps the user’s identity to a local identity, creates an entry in the server cache, performs an access control decision and provides the appropriate response to the user’s request. This front-end is now able to vouch for this user’s identity within the domain.

Process for Transferring Vouch-For Information Across Domains

[0094] Vouch for information is transferred across domains when a user requests a resource in a domain other than their home domain, where the request requires an authenticated identity. Referring again to the components of Figure 1, the overall transferring vouch for information across domains is described as follows, assuming that the user does not have an eCC cookie for this front-end and that the user already has DIDC cookie for this domain:

- (1) The user requests access to a resource protected by a plug-in at an associated domain (106). Included in this request is DIDC for the associated domain, such as:

DIDC(106) = {home domain = 103,
vouch for URI = www.103.com/101/vouch_for.htm,
e-community = sample,
creation date = Nov 1, 2000}, hash(info).

- (2) The plug-in at the associated domain front-end (104) determines

that it does not have a current, vouched-for or authenticated session for this user (100). This could be based on the presence or absence of an eCC cookie set by the associated domain front-end (104), or some other means, such as SSL session ID mapping as used by Policy Director with secure hypertext transfer protocol (HTTPS).

- (3) The plug-in at the associated domain front-end (104) looks for an eCC cookie set by a different front-end within the associated domain (106). If present, this would indicate that the user has a session with a different front-end within the associated domain (106).

- (4) The plug-in then “parses” the DIDC(106) cookie to determine the user’s home domain, a URI in their home domain that can vouch for the user’s identity, the e-community in which they are enrolled, and a creation/update timestamp.

- (5) The plug-in also verifies (from the timestamp) that the user’s enrollment has not expired. If the user’s enrollment has expired, plug-in will respond to the user with an “enrollment expired, please re-enroll in e-community X” message and the processing will stop.

- (6) Otherwise, the plug-in generates a response to the user, which is re-directed to the home domain and requesting front-end (101), requesting that domain 103 vouch for the user's identity.

- (7) The user's browser (100) then re-directes the response from the

affiliated domain front-end (104) to the home domain front-end (101),
including the home domain DIDC.

(8) The home domain plug-in at the first front-end (101) receives the
“vouch-for” request from the affiliated domain front-end (104) via
the user's browser (100), and determines if the user has a currently
authenticated session within the home domain (103). If not, the
plug-in initiates an authentication process, as previously described.

(9) The home domain plug-in at the first front-end (101) updates the
timestamp in user eCC from the first front-end (101).

(10) Based on the authenticated identity, the first front-end (101) builds a
“Vouch-For Token” (VT) to provide the vouch-for information to
the affiliated domain (106), such as:

$$\text{VT} = E\{\text{Tag} = \text{VT}, \text{userid} = \text{jsmith}, \text{home domain} = 103, \\ \text{e-community} = \text{sample}, \text{timestamp} = 00.01:01, \text{extensible} \\ \text{attribute}=\text{value pairs} \}$$

where the information is encrypted, $E\{\text{---}\}$.

(11) The home domain plug-in constructs a response to the affiliated
domain front-end (104), which is sent via redirection through the
user's browser (100) with the VT appended to the URI. The updated
eCC is included in this response.

(12) The user's browser (100) extracts the eCC cookie and puts it in its
memory store, and redirectes (e.g. forwards) the message on to

the affiliated domain's front-end (104).

(13) The affiliated domain's front-end (104) plug-in (109) receives the re-directed response originating from the home domain, and extracts the VT from the URI.

5 (14) The affiliated domain plug-in then extracts the information in the VT, and verifies the timestamp. If the timestamp is not "fresh", there may have been an unnecessary delay in transferring the information, so the affiliated domain plug-in may reinitiate the request to vouch for identity.

10 (15) Based on the user's identity and home domain, and optionally the extensible data, the affiliated domain plug-in maps the user to a username that is valid within the affiliated domain (106), and the affiliated domain front-end (104) then creates a credential for the user for use in subsequent access control decisions.

15 (16) The affiliated domain plug-in then creates an eCC cookie for the user, such as:

eCC(104) = {Auth Server = 104,

URI at Auth Server = www.106.com/104/vouch_for.htm,

e-community = sample,

20 creation date = Nov. 1, 2000,

extensible attribute=value pairs}, hash(info)

(17) The affiliated domain plug-in performs an access control decision on

the protected resource given the user's identity and credential.

(18) The affiliated domain plug-in responds to the user's browser (100) based on the results of the access control decision, including the eCC for the affiliated domain front-end (104).

5 (19) The user's browser (100) receives the response and stores the eCC for the affiliated domain's front-end (104) in its cookie store.

As a result, the following conditions are established:

a. The user's browser (100) has a memory single-sign-on cookie set by the affiliated domain's front-end (104), such as:

10 eCC(104) = {Auth Server = 104, URI at Auth Server =
www.106.com/104/vouch_for.htm, e-community = sample,
creation date = Nov. 1, 2000, extensible attribute=value
pairs}, hash(info)

b. The user is also authenticated to his home domain (103).

15 Authorization in the new CD-SSO Domain

[0095] When a user has successfully transferred to a participating e-community domain, authorization is performed as previously described, allowing a domain-specific user identity to be established as part of the cross-domain

single-sign-on. Based on this user id, the web front-end can retrieve the user's credentials and perform the normal access control decision.

Logout of the e-community

[0096] According to the preferred embodiment, a user can be "logged out" of an e-community session in one of several ways. A user can explicitly logout, the user can end their browser session, or the user can logout by virtue of an inactivity timeout. As a result of the logout procedure, the user's e-community cookies and any other session state management "records" are invalidated.

[0097] In the first method of log-out, when the user ends all active browser sessions, such by closing their web browser software, the user's host cookies including all e-community cookies are invalidated and any active SSL sessions are terminated. This means that the e-community SSO functionality is no longer usable - the user must re-authenticate to the e-community. This form of logout requires no changes or modifications to the policy server or the e-community functionality.

[0098] The disadvantage of this type of logout is that it is not "recorded" at the server side, and the e-community plug-in does not know that the user has terminated their session. If a user's session did not occur over a protected link, then it would be possible for an attacker to steal the user's e-community cookie and replay it to "establish" a session with the server. This is one reason why we recommend that all communications occur over SSL or a similar protected link where possible.

[0099] In the second method of log-out, inactivity logout may occur when a user's SSL session expires when using a security policy server which supports session timeouts. The duration of an SSL session is browser dependent, but generally configured to eight hours. This type of "logout" will not lead to a complete logout of the e-community but will provide a "staggered" logout, where the user is logged out of each individual domain as his SSL session with that domain expires.

[0100] If the eCC cookie is being used for state management, inactivity logout will occur when the lifetime as dated from the creation date of the e-community cookie expires. This approach must deal with clock-skew across replicas and clusters within a domain.

[0101] The only way to get an inactivity logout based on the e-community cookie would be to put a timestamp in the eCC and to require the front-end to check this timestamp, regardless of the presence of a valid SSL connection.

[0102] According to the third method of log-out, if the user invokes an explicit logout command, the e-community plug-in may "kill" the existing SSL session and all SSL sessions within other e-community domains, as well as expire the e-community generated e-community cookies.

[0103] The logout functionality is preferably maintained in the "e-community portal", and may be integrated with the enrollment functionality. The home domain preferably redirects logout messages to all e-community domains in which the user has enrolled based on the user profile stored in their domain identity cookie(s). It is the

responsibility of these other domains to expire the user's e-community cookies and to handle the log-out functionality within their domain.

Key Management Issues

[0104] The present e-community single-sign-on invention requires one set of "keys",
5 a set of symmetric keys used to protect the tokens such as vouch-for, enrollment and
logout tokens.

[0105] These keys are preferably maintained in an extensible markup language
("XML") table. This data can, at its finest level of granularity, be maintained pairwise
between all members of the e-community. In basic embodiment, there may be only one
10 set of keys shared by all members of the e-community. According to the preferred
embodiment using a standard web security policy server, a table will be provided that
can be populated on a pairwise basis, initialized with universally shared keys.

[0106] Further according to the preferred embodiment, all re-keying should be done
manually, including the regeneration and distribution to all affiliates of the shared
15 keying material. This approach allows for periodic re-keying of the encryption keys
used. This is configurable for the installation, and will apply to the entire
e-community. This can be triggered when a member joins or leaves the e-community
or if for some reason any member has reason to request a re-key.

Cookies and Tokens

[0107] The invention uses tokens and HTTP cookies. Tokens are transitory blocks of data used to communicate across domains, while cookies follow well-known HTTP specifications and are used to store information for a given domain.

- 5 [0108] Tokens are preferably protected cryptographically using encryption, using HTTPS or other suitable means.

[0109] A symmetric shared key is used for the encryption of the tokens exchanged. It is recommended that this key be updated on a regular, frequent basis. As such, this implementation should facilitate the updating of shared keying material. The

- 10 CryptoLite package from IBM-Zurich may be used to provide the encryption algorithms for this functionality.

[0110] Tokens are used to pass information across DNS domains. They are appended to the URI in the HTTP request/response message and are preferably limited to 2Kbyte according to the preferred embodiment in order to meet industry standards
15 for URI size limitations in the HTTP header.

[0111] The enrollment token is used to introduce a user's identity to participating e-community domains, as previously described. The enrollment token contains the following data items:

- a. An ET tag: This is used to identify the token as an ET token.
- 20 b. The user's home domain: Together with the user's identity as received in the vouch-for token, this is used to map the user to an identity that is valid within the new domain.

- c. The e-community identity: This is used to identify which e-community the user is participating in.
- d. A vouch-for resource: This is a URL that contains some form of active content that can authenticate the user (if necessary) and build a “vouch for” token.
- e. A timestamp: This is used to limit the lifetime of the enrollment token. The lifetime is configurable so that different implementations can allow for enrollment tokens with varying lifetimes.
- f. Extensible data: Attribute-value pairs which may be used by the “introduced to” domain.
- g. Hashed information: All of the preceding information a - f is preferably signed with a keyed hash for non-repudiation purposes. This data is also limited in size. The entire hashed, encrypted ET token, together with the redirected URI cannot exceed 2 Kbytes.
- 15 **[0112]** A sample enrollment token for user Jane Doe may look like the following:
- Enrollment token:= Tag = ET,
- HomeDomain = ibm.com, e-communityIdentity = f3Closed,
- VouchFor = www.ibm.com/vouchfor.htm,
- timestamp = 11:10:00 2 Nov 2000,
- 20 group = accountant}, hash(info)

[0113] A refresh enrollment token is the same as the enrollment token, with a “refresh enrollment” tag instead of an “enrollment” tag. A refresh enrollment token is used to indicate that a refresh action is required on a user’s existing identity cookie. A SSO plug-in takes the user’s domain identity cookie and refreshes the cookie

5 timestamp based on the correlation of information in the re-enrollment token and the user's domain identity cookie. If a refresh enrollment token is received and the user does not have a domain identity cookie in that domain, the SSO plug-in may create a new enrollment token based on the information in the refresh enrollment token.

[0114] A vouch-for token ("VT") is used to vouch for the identity of an already
10 authenticated user to a domain other than the user’s home domain. The token contains the following data items:

- a. VT tag: This is used to identify the token as a VT token.
- b. The user’s identity: Together with the user’s home domain this is used to map the user to an identity that is valid within the new domain.
- 15 c. The user’s home domain: Together with the user’s identity this is used to map the user to an identity that is valid within the new domain.
- d. The e-community identity: This is used to identify which e-community the user is participating in.
- e. Extensible data in the form of attribute-value pairs may be used by the
20 “introduced to” domain. This data is also limited in size: the entire

hashed, encrypted VT token, together with the redirected URI cannot exceed 2Kbytes.

- f. A timestamp: This is used to limit the lifetime of the enrollment token. The lifetime is configurable so that different implementations can allow for enrollment tokens with varying lifetimes.

[0115] A same vouch for token for user Jane Doe may look like the following:

Vouch For Token := E{Tag = VT, UserIdentity = jjdoe,
HomeDomain = ibm.com,
e-communityIdentity = f3Closed, timestamp = 11:10:00 2 Nov 2000,
group=accountant,
role=manager }

[0116] Cookies are preferably used to maintain persistent data between a user and a given domain. The persistent identity cookies used in the preferred embodiment do not contain any security relevant information, and simple possession of a domain identity cookie does not provide access to a system.

[0117] The e-community memory cookies contain security relevant information such that possession of an e-community cookie may provide access to a particular session. This situation arises if the e-community cookie is being used for state management (for unprotected HTTP communications). This implies that the e-community cookies should be protected against theft.

[0118] A domain identity cookie ("DIDC") is a persistent cookie that resides in the user's cookie "jar", such as a cookie.txt file. This cookie is used to identify the e-community in which the user has enrolled, and their e-community in which the user is participating, and a timestamp. The identity cookie may also include extensible data in the form of attribute=value pairs.

This information is used as follows:

- a. The e-community identity is used to identify which e-community the user is participating in.
- b. The vouch for resource will be a URL that contains some form of active content that can authenticate the user (if necessary) and build a "vouch for" token.
- c. Extensible data in the form of attribute-value pairs may be used by the "introduced to" domain. This data will not be signed or encrypted (see note below). This data is also limited in size such that the entire domain identity cookie cannot exceed 4Kbytes, according to the preferred embodiment.
- d. The timestamp is used to limit the lifetime of the enrollment token. The lifetime is configurable so that different implementations can allow for enrollment tokens with varying lifetimes.
- e. The information is hashed for integrity protection.

[0119] A sample domain identity cookie from domain "sun.com" for user Jane Doe may look like the following:

5 DIDC(sun.com) := {Home Domain = ibm.com,
 e-communityIdentity = f3Closed,
 timestamp = 11:10:00 2 Nov 2000,
 branding = ibm_sun alliance}, hash(info)

[0120] According to the preferred embodiment, the information in the domain identity cookie is not be protected by encryption. The domain identity cookie is intended to a piece of information for long-term use. Thus, if it were encrypted, the
10 key management issues would be complicated beyond the benefits of this information. So, the domain identity cookie may or may not be protected by keyed hash. The keyed hash protection will at most provide integrity protection on the data in the cookie. Thus, the data that is included in the cookie should not be security relevant, but should be limited to information used for branding and personalization purposes.
15 As possession of an identity cookie by itself is not enough to provide access to a system, it does not provide a potential attacker with any form of advantage to steal such a cookie.

[0121] The DIDC cookie is timestamp "refreshed" every time a user authenticates or is vouched-for within a domain within the e-community. This cookie is also refreshed
20 as a result of a refresh request.

[0122] Note that the structure of the domain identity cookie as set by the e-community home domain is slightly different. In the extensible data field there is a list or a pointer to a list of all the individual affiliate sites where the user has explicitly enrolled.

5 [0123] Turning now to the e-community cookie (eCC), it resides in the user's browser cookie memory. The eCC is used to identify the Web server cluster, within a given domain, that can vouch for a user's identity, and is not encrypted.

[0124] The eCC has a timestamp and optionally extensible data, as well, which is refreshed upon every request to the Web server containing the e-community SSO
10 plug-in. An eCC is valid for only for the duration of a browser session, and it is expired when a user invokes logout functionality.

[0125] The e-community cookie contains the following data items:

- a. A local vouch for Web server: this will identify which web server (within a cluster or within the domain) has received the vouch for
15 information from the user's home domain.
- b. A local vouch for resource: this will be a URL that contains some form of active content that can vouch for the user within the scope of the local domain.
- c. Extensible data: Attribute-value pairs may be included.
- 20 d. A timestamp: this is used to limit the lifetime of the enrollment token. The lifetime is configurable so that different implementations can allow for enrollment tokens with varying lifetimes.

- e. Hash information: This data will be signed with a keyed hash for non-repudiation purposes. This data is also limited in size: the entire identity cookie cannot exceed 2 Kbytes.

[0126] A sample e-community cookie from domain "sun.com" for user Jane Doe

5 appears as such:

eCC(sun.com) := {LocalVouchFor = sol.sun.com,
LocalVouch Resource = www.sol.sun.com/f3vouch.html,
eCommunity ID = f3Closed, timestamp = 11:1000 2 Nov 2000,
group=accountant, role=manager, role=approver, approval
10 limit=\$50,000}, hash(info)

Non-Secured Implementation of e-Community SSO

[0127] Not all implementations of an e-community will require strong protection on e-community communications. An example of this type of implementation would be a B2C environment or a similar environment that did not have legal and liability
15 implications attached to the misuse of a user's identity.

[0128] For these situations, it is possible to configure the e-community functionality to not implement cryptographic protection on the cookies and tokens used to maintain and transfer information within the e-community. They will be required to maintain some form of active content for populating metadata to the cross-domain tokens.

[0129] The vouch-for tokens are normally all encrypted in the secure process. In the non-secured implementation of the e-community process, they are not confidentiality or integrity protected.

[0130] A sample enrollment token in such a case may appear as:

5 Enrollment Token := Tag = ET,

 HomeDomain = aol.com,

 e-communityIdentity = aolShopping,

 VouchFor = www.aol.com/vouchfor.htm,

 timestamp = 11:10:00 1 Nov 2000,

10 group = consumer

[0131] Further, a sample vouch-for token in such a case may take the form of:

 Vouch For Token : = Tag=VT,

 HomeDomain = aol.com,

 e-communityIdentity = aolShopping,

15 timestamp = 15:10:00 2 Nov 2000

[0132] These tokens should be used in unprotected format only when there is no risk associated with the theft and replay of these tokens. Neither the eCC or DIDC cookie is normally confidentiality or integrity protected in the secured process, and as such, there are no changes required for the unsecured version of this process.

[0133] While a preferred embodiment has been disclosed along with aspects of alternate embodiments, it will be recognized by those skilled in the art that certain variations and substitutions from the methods, systems and arrangements disclosed will not depart from the spirit and scope of the invention. Therefore, the scope of the

5 present invention should be determined by the following claims.